Calculate Semester Grade

Five Individual Arrays

```
char id[ 4 ][ 8 ];
char name[ 4 ][ 8 ];
int homework[ 4 ];
int midterm[ 4 ];
int final[ 4 ];
```

id[0]	932201
id[1]	932202
id[2]	932203
id[3]	932204

name[0]	Chen
name[1]	Lee
name[2]	Lin
name[3]	Wu

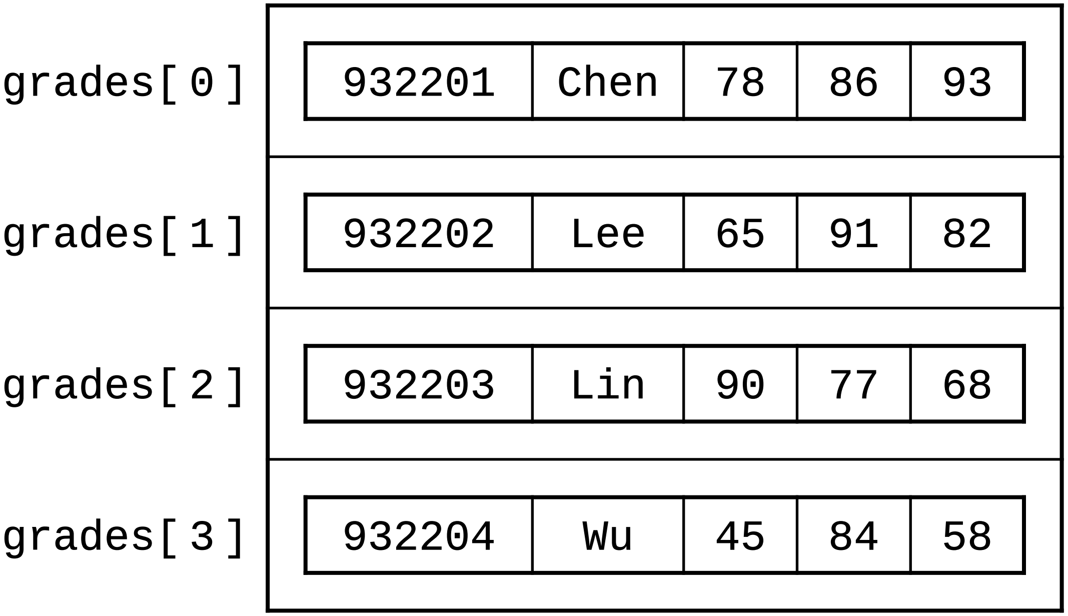
homework[0]	78
homework[1]	65
homework[2]	90
homework[3]	45

midterm[0]	86
midterm[1]	91
midterm[2]	77
midterm[3]	84

final[0]	93
final[1]	82
final[2]	68
final[3]	58

Array of struct

```
struct Grade
{
   char id[ 8 ];
   char name[ 8 ];
   int midTerm;
   int final;
   int homework;
};

Grade grades[ 4 ];
```



grades[0]	932201	Chen	78	86	93
grades[1]	932202	Lee	65	91	82
grades[2]	932203	Lin	90	77	68
grades[3]	932204	Wu	45	84	58

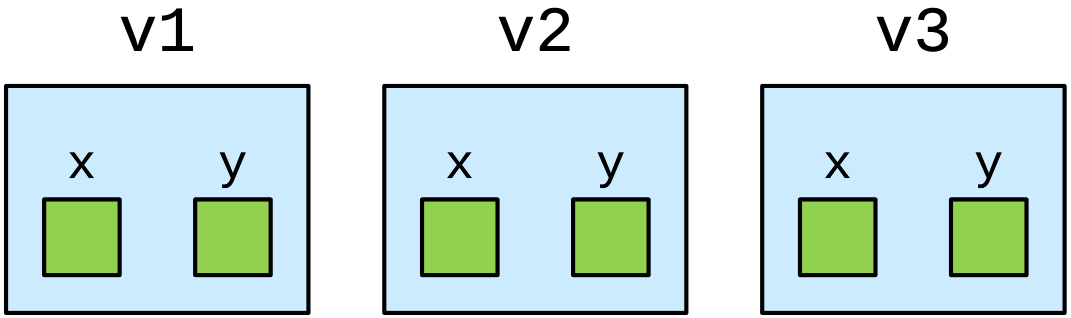
```
int main()
{
   const int numStudents = 2;
   char id[ numStudents ][ 8 ];
   char name[ numStudents ][ 8 ];
   int homework[ numStudents ];
   int midTerm[ numStudents ];
   int final[ numStudents ];
   for( int i = 0; i < numStudents; i++ )
   {
      cout << "ID: ";
      cin >> id[i];
      cout << "name: ";
      cin >> name[i];
      cout << "homework grade: ";
      cin >> homework[i];
      cout << "midterm exam grade: ";
      cin >> midTerm[i];
      cout << "final exam grade: ";
      cin >> final[i];
      cout << endl;
   }
   for( int i = 0; i < numStudents; i++ )
      cout << "The semester grade of " << name[i] << " is "
           << ( midTerm[i] + homework[i] ) * 0.2 + final[i] * 0.6
           << endl;
}
```

```cpp
int main()
{
   int i;
   const int numStudents = 2;
   Grade grades[ numStudents ];
   for( i = 0; i < numStudents; i++ )
   {
      cout << "ID: ";
      cin >> grades[i].id;
      cout << "name: ";
      cin >> grades[i].name;
      cout << "midterm exam grade: ";
      cin >> grades[i].midTerm;
      cout << "final exam grade: ";
      cin >> grades[i].final;
      cout << "homework grade: ";
      cin >> grades[i].homework;
      cout << endl;
   }
   for( i = 0; i < numStudents; i++ )
      cout << "The semester grade of " << grades[i].name << " is "
           << ( grades[i].homework + grades[i].midTerm ) * 0.2 +
                grades[i].final * 0.6 << endl;
}
```

Vectors

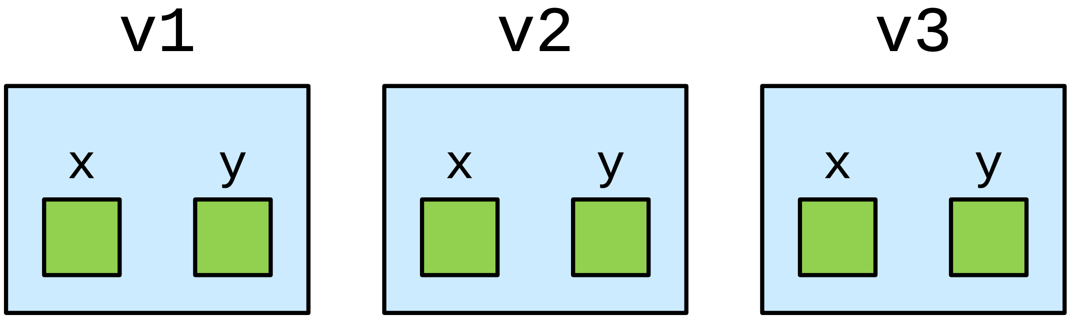
```
void add( int u1_x, int u1_y, int u2_x, int u2_y, int &u3_x, int &u3_y );
double length( int u_x, int u_y );

int main()
{
   int v1_x, v1_y, v2_x, v2_y, v3_x, v3_y;
   v1_x = rand() % 10;
   v1_y = rand() % 10;
   v2_x = rand() % 10;
   v2_y = rand() % 10;

   add( v1_x, v1_y, v2_x, v2_y, v3_x, v3_y );
   cout << length( v3_x, v3_y ) << endl;
}

void add( int u1_x, int u1_y, int u2_x, int u2_y, int &u3_x, int &u3_y )
{
   u3_x = u1_x + u2_x;
   u3_y = u1_y + u2_y ;
}

double length( int u_x, int u_y )
{
   return sqrt( static_cast< double >( u_x * u_x + u_y * u_y ) );
}
```

```
struct Vector
{
   int x;
   int y;
};

void add( Vector u1, Vector u2, Vector &u3 );
double length( Vector u );

int main()
{
   Vector v1, v2, v3;
   v1.x = rand() % 10;
   v1.y = rand() % 10;
   v2.x = rand() % 10;
   v2.y = rand() % 10;
   add( v1, v2, v3 );
   cout << length( v3 ) << endl;
}
```

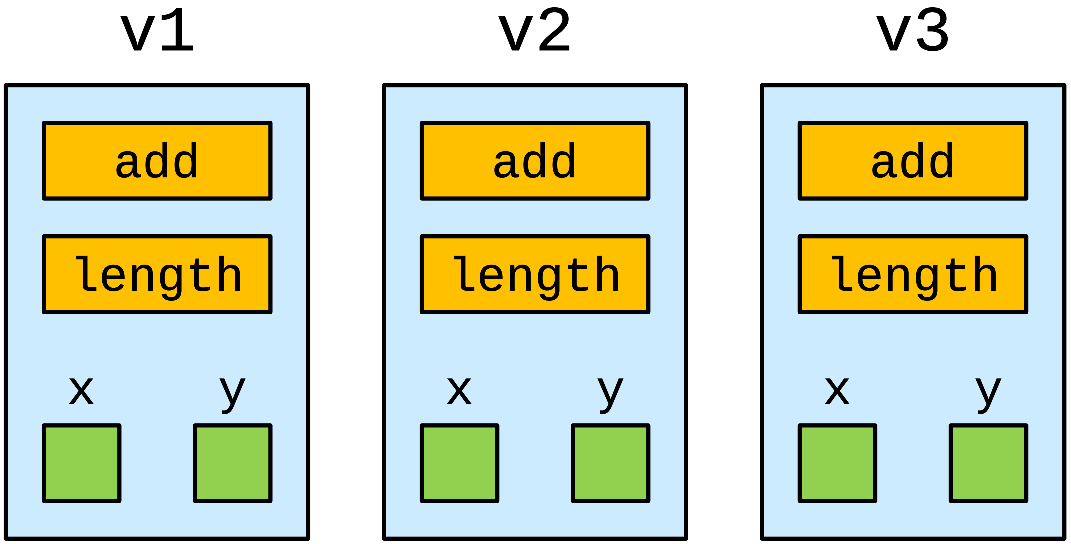


```
void add( Vector u1, Vector u2, Vector &u3 )
{
   u3.x = u1.x + u2.x;
   u3.y = u1.y + u2.y;
}
```

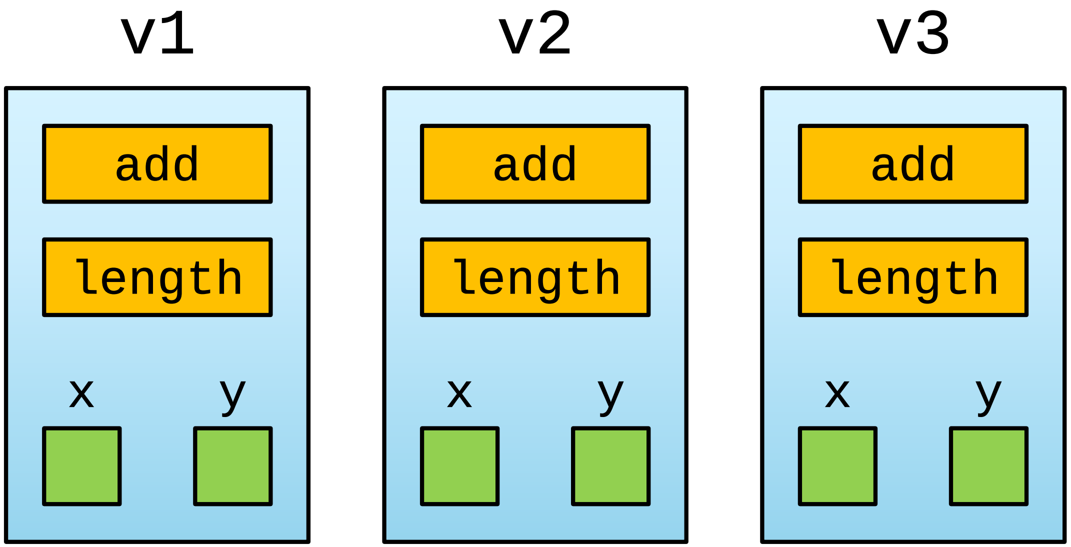
```
struct Vector
{
   int x;
   int y;
};

Vector add( Vector u1, Vector u2 );
double length( Vector u );

int main()
{
   Vector v1, v2, v3;
   v1.x = rand() % 10;
   v1.y = rand() % 10;
   v2.x = rand() % 10;
   v2.y = rand() % 10;
   v3 = add( v1, v2 );
   cout << length( v3 ) << endl;
}
Vector add( Vector u1, Vector u2 )
{
   Vector u3;
   u3.x = u1.x + u2.x;
   u3.y = u2.y + u2.y;
   return u3;
}
```



```
class Vector
{
public:
   Vector add( Vector u2 );
   double length();
   int x;
   int y;
};

int main()
{
   Vector v1, v2, v3;
   v1.x = rand() % 10;
   v1.y = rand() % 10;
   v2.x = rand() % 10;
   v2.y = rand() % 10;
   v3 = v1.add( v2 );
   cout << v3.length() << endl;
}
Vector Vector::add( Vector u2 )
{
   Vector u3;
   u3.x = x + u2.x;
   u3.y = y + u2.y;
   return u3;
}
```



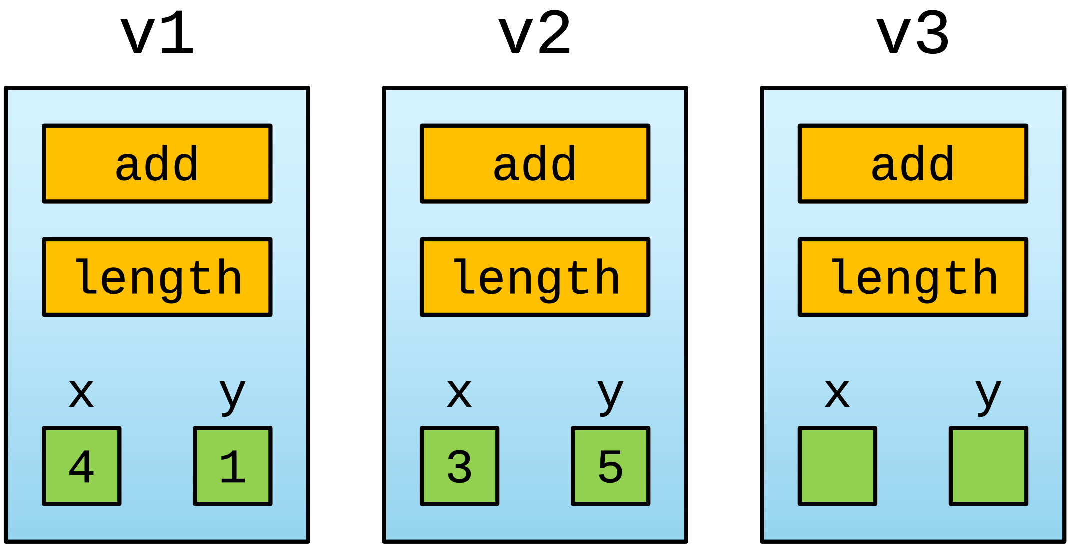
```
int main()
{
   Vector v1, v2, v3;
   v1.x = rand() % 10;
   v1.y = rand() % 10;
   v2.x = rand() % 10;
   v2.y = rand() % 10;
   v3 = v1.add( v2 );
   cout << v3.length();
}
```



```
double Vector::length()
{
   return sqrt( x*x + y*y );
}
```

```
Vector Vector::add( Vector u2 )
{
   Vector u3;
   u3.x = x + u2.x;
   u3.y = y + u2.y;
   return u3;
}
```

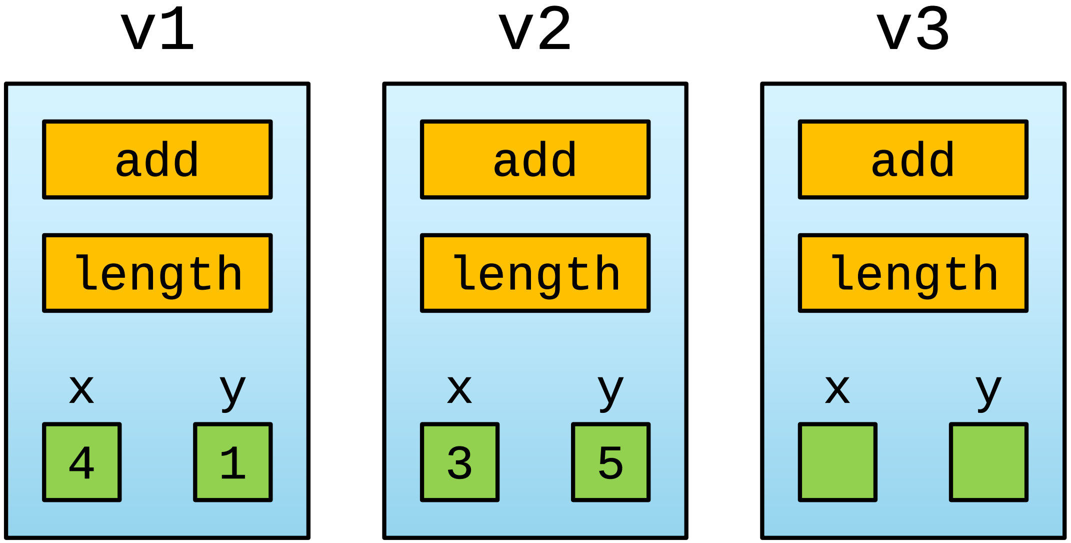
```
int main()
{
   Vector v1, v2, v3;
   v1.x = rand() % 10;
   v1.y = rand() % 10;
   v2.x = rand() % 10;
   v2.y = rand() % 10;
   v3 = v1.add( v2 );
   cout << v3.length();
}
```



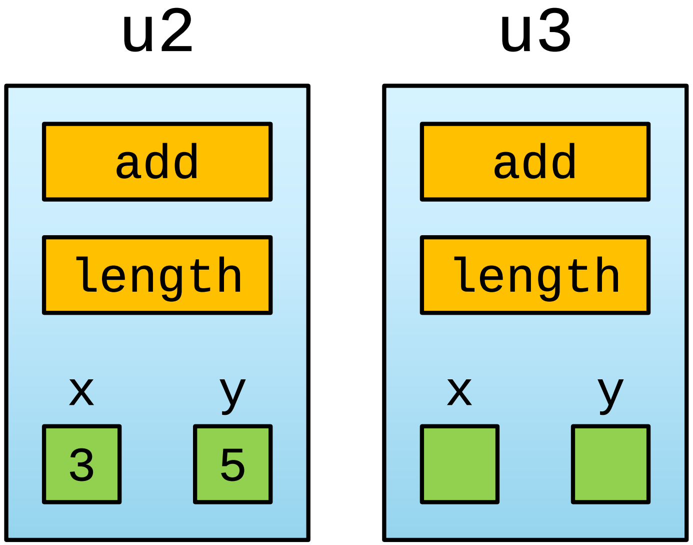
```
double Vector::length()
{
   return sqrt( x*x + y*y );
}
```

```
Vector Vector::add( Vector u2 )
{
   Vector u3;
   u3.x = x + u2.x;
   u3.y = y + u2.y;
   return u3;
}
```

```
int main()
{
   Vector v1, v2, v3;
   v1.x = rand() % 10;
   v1.y = rand() % 10;
   v2.x = rand() % 10;
   v2.y = rand() % 10;
   v3 = v1.add( v2 );
   cout << v3.length();
}
```

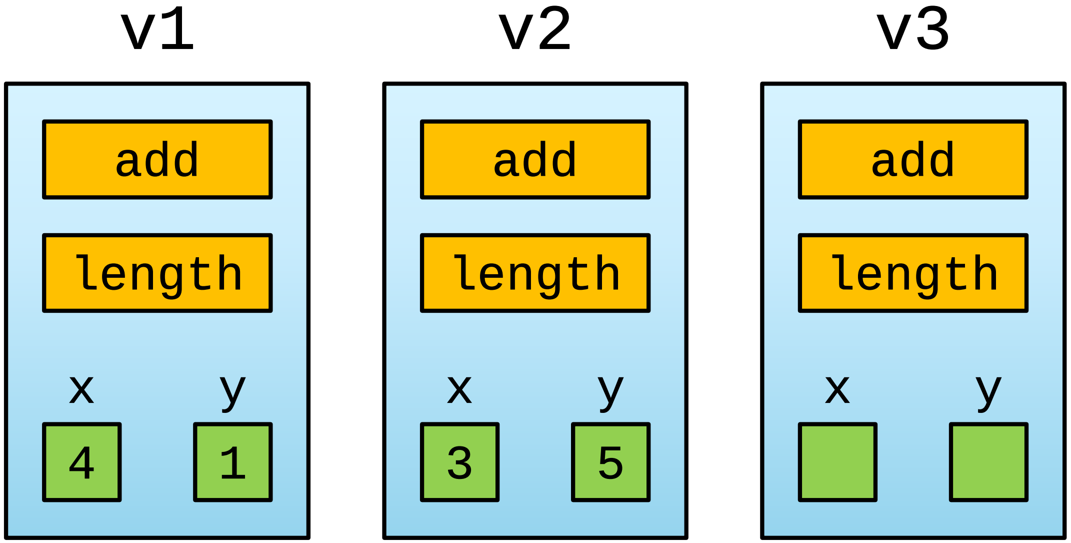


```
double Vector::length()
{
   return sqrt( x*x + y*y );
}
```



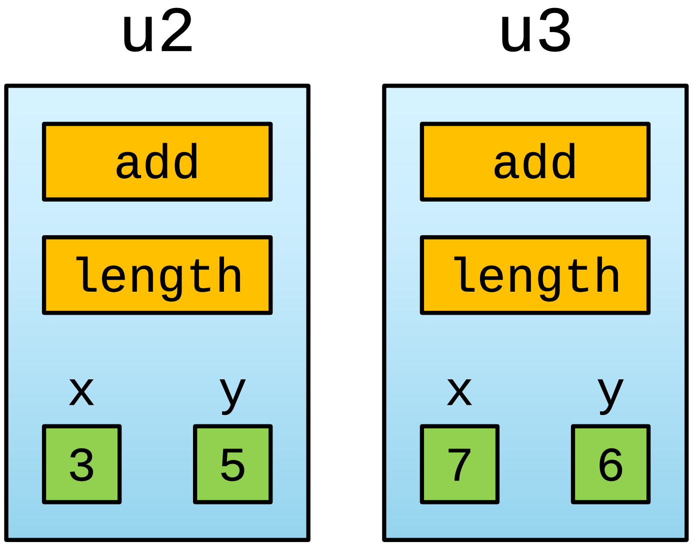
```
Vector Vector::add( Vector u2 )
{
   Vector u3;
   u3.x = x + u2.x;
   u3.y = y + u2.y;
   return u3;
}
```

```
int main()
{
   Vector v1, v2, v3;
   v1.x = rand() % 10;
   v1.y = rand() % 10;
   v2.x = rand() % 10;
   v2.y = rand() % 10;
   v3 = v1.add( v2 );
   cout << v3.length();
}
```

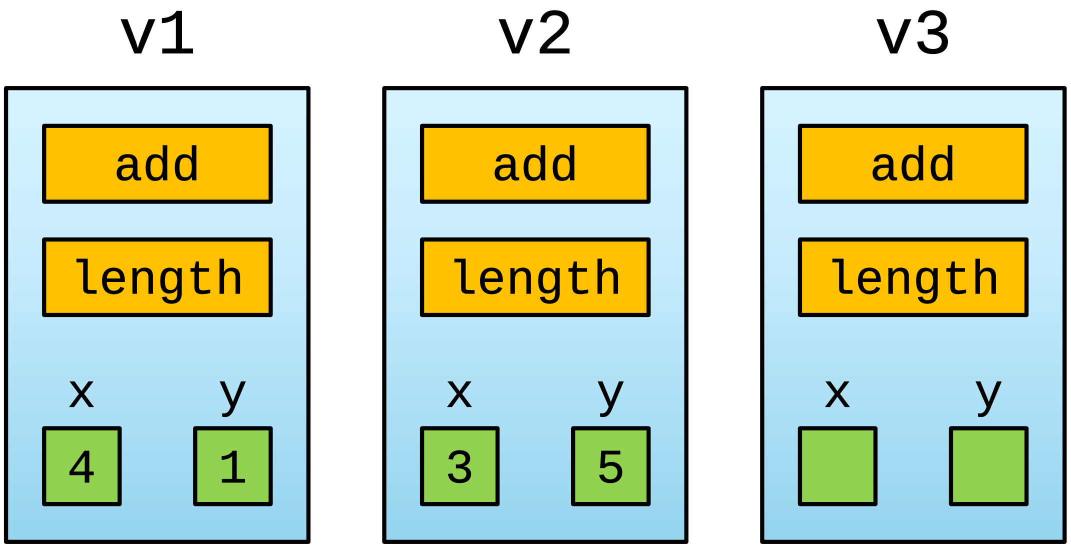


```
double Vector::length()
{
   return sqrt( x*x + y*y );
}
```

```
Vector Vector::add( Vector u2 )
{
   Vector u3;
   u3.x = x + u2.x;
   u3.y = y + u2.y;
   return u3;
}
```

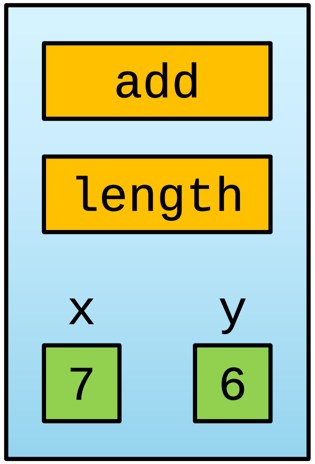


```
int main()
{
   Vector v1, v2, v3;
   v1.x = rand() % 10;
   v1.y = rand() % 10;
   v2.x = rand() % 10;
   v2.y = rand() % 10;
   v3 = v1.add( v2 );
   cout << v3.length();
}
```

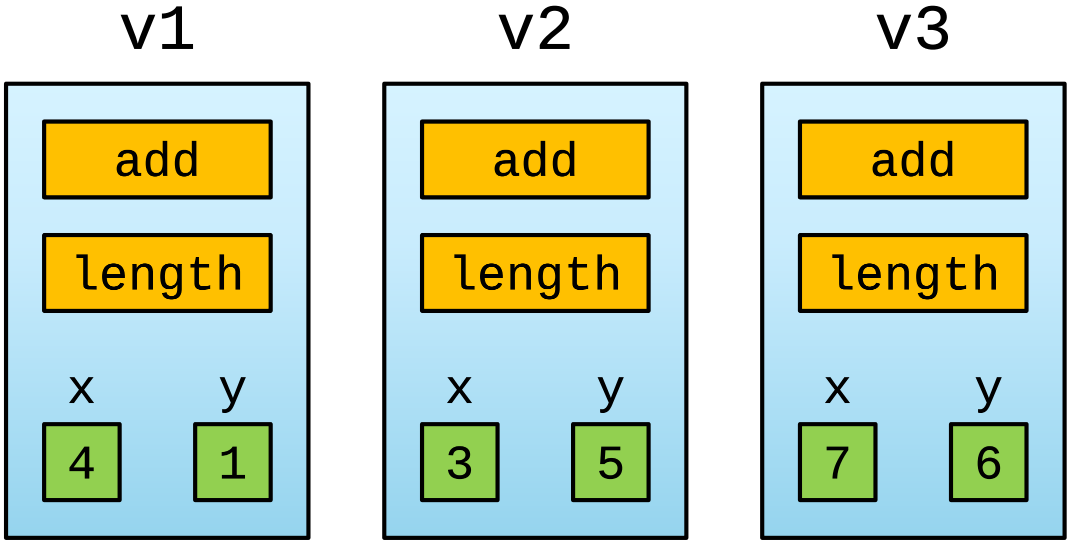


```
double Vector::length()
{
   return sqrt( x*x + y*y );
}
```

```
Vector Vector::add( Vector u2 )
{
   Vector u3;
   u3.x = x + u2.x;
   u3.y = y + u2.y;
   return u3;
}
```

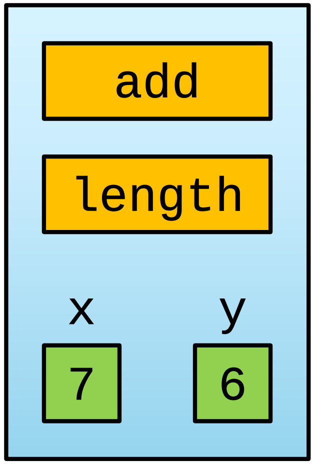


```
int main()
{
   Vector v1, v2, v3;
   v1.x = rand() % 10;
   v1.y = rand() % 10;
   v2.x = rand() % 10;
   v2.y = rand() % 10;
   v3 = v1.add( v2 );
   cout << v3.length();
}
```

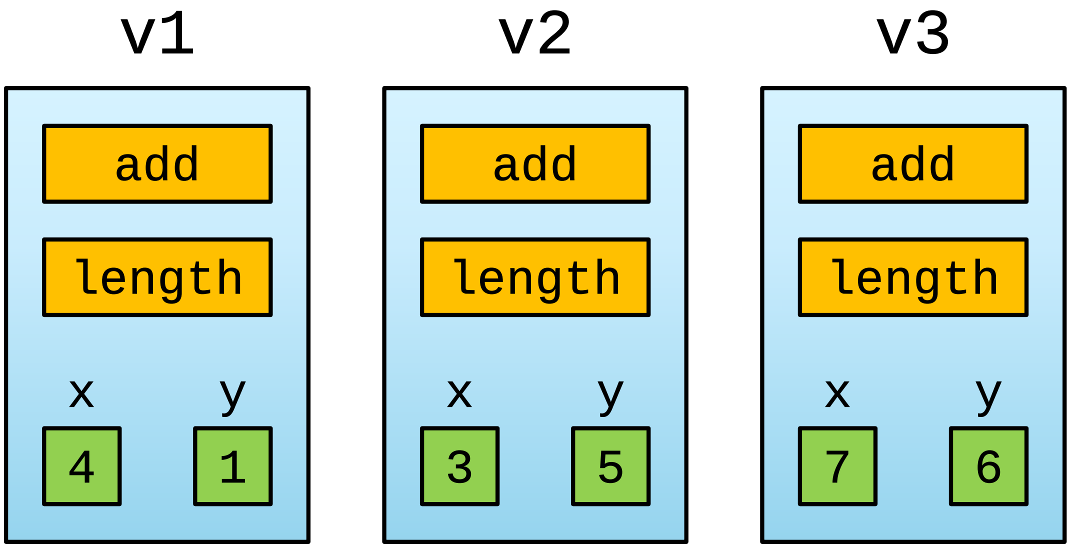


```
double Vector::length()
{
   return sqrt( x*x + y*y );
}
```

```
Vector Vector::add( Vector u2 )
{
   Vector u3;
   u3.x = x + u2.x;
   u3.y = y + u2.y;
   return u3;
}
```



```
int main()
{
   Vector v1, v2, v3;
   v1.x = rand() % 10;
   v1.y = rand() % 10;
   v2.x = rand() % 10;
   v2.y = rand() % 10;
   v3 = v1.add( v2 );
   cout << v3.length();
}
```



```
double Vector::length()
{
   return sqrt( x*x + y*y );
}
```

```
Vector Vector::add( Vector u2 )
{
   Vector u3;
   u3.x = x + u2.x;
   u3.y = y + u2.y;
   return u3;
}
```